

## Задача А. Расстановка скобок в лямбда-выражении

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 512 мегабайт

На вход вашей программе дается лямбда-выражение в следующей грамматике:

$$\begin{aligned}\langle \text{Выражение} \rangle &::= [\langle \text{Применение} \rangle] \backslash \langle \text{Переменная} \rangle \text{'.'} \langle \text{Выражение} \rangle \\ &\quad | \langle \text{Применение} \rangle \\ \langle \text{Применение} \rangle &::= \langle \text{Применение} \rangle \langle \text{Атом} \rangle | \langle \text{Атом} \rangle \\ \langle \text{Атом} \rangle &::= \text{'('} \langle \text{Выражение} \rangle \text{'')}' | \langle \text{Переменная} \rangle \\ \langle \text{Переменная} \rangle &::= (\text{'a'... 'z'}) \{ \text{'a'... 'z'} | \text{'0'... '9'} | \text{'.'} \}^*\end{aligned}$$

Аргументы-переменные в применении должны разделяться пробелом. В остальных случаях пробелы могут отсутствовать. Любые пробелы между нетерминальными символами (кроме пробела, разделяющего аргументы в применении) — а также начальные и конечные пробелы в строке — должны игнорироваться. Символы табуляции, возврата каретки и перевода строки должны трактоваться как пробелы.

Требуется расставить все недостающие скобки вокруг всех абстракций и применений, и напечатать получившийся результат.

### Формат входных данных

Размер входного файла не превышает 1 МБ.

### Формат выходных данных

В единственной строке выходного файла (заканчивающейся переводом строки) должно быть приведено лямбда-выражение с расставленными скобками без каких-либо пробельных символов. Исключение: одиночные пробелы, разделяющие аргументы в применении.

### Примеры

стандартный ввод
<code>\a.\b.a b c (\d.e \f.g) h</code>
стандартный вывод
<code>(\a.(\b.((((a b) c) (\d.(e (\f.g)))) h)))</code>
стандартный ввод
<code>((a\bbb.c)d)e f g</code>
стандартный вывод
<code>(((((a (\bbb.c)) d) e) f) g)</code>

## Задача В. Нормализация лямбда-выражения

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	1024 мегабайта

Дано лямбда-выражение, требуется провести  $m$  ( $m \in \mathbb{N}_0$ ) бета-редукций этого выражения используя нормальный порядок редукции и мемоизацию, при этом выводить на печать требуется каждое  $k$ -е выражение ( $k \in \mathbb{N}_0, k < m$ ). Формулы нумеруются с 0, если нормальная форма была достигнута на формуле с некрatным  $k$  номером — на формуле  $\delta_s$ , где  $k \cdot (n-1) < s < k \cdot n$ , — то выдача должна завершиться формулой  $\delta_s$ . Например, редуцирование выражения  $(\lambda x.x\ x\ x\ x)\ ((\lambda x.x)\ (\lambda x.x))$  в данных условиях пройдёт через следующие стадии (редуцируемые бета-редексы подчёркнуты):

обозначение (номер)	формула
$\delta_0$	$(\lambda x.x\ x\ x\ x)\ ((\lambda x.x)\ (\lambda x.x))$
$\delta_1$	$((\lambda x.x)\ (\lambda x.x))\ ((\lambda x.x)\ (\lambda x.x))\ ((\lambda x.x)\ (\lambda x.x))\ ((\lambda x.x)\ (\lambda x.x))$
$\delta_2$	$(\lambda x.x)\ (\lambda x.x)\ (\lambda x.x)\ (\lambda x.x)$
$\delta_3$	$(\lambda x.x)\ (\lambda x.x)\ (\lambda x.x)$
$\delta_4$	$(\lambda x.x)\ (\lambda x.x)$
$\delta_5$	$(\lambda x.x)$

Если при этом  $k = 2$ , то на печать должны быть выведены формулы  $\delta_0, \delta_2, \delta_4, \delta_5$ .

Гарантируется, что суммарная длина всех выражений, которые будут получены в результате  $s$  бета-редукций, не превышает 100 миллионов лексем.

Для точного определения условий задачи, давайте напомним два важных определения — нормальный порядок редукций и мемоизацию.

1. Рассмотрим лямбда-выражение, расставим все необязательные скобки в нём. Назовём нормальным порядком редукции такой порядок, при котором всегда редуцируется самый левый редекс: то есть редекс, первый символ которого находится левее всего в выражении.
2. Чтобы определить мемоизацию, определим некоторое расширенное лямбда-исчисление. Помимо обычных выражений будем рассматривать отложенные подстановки: это переменные с указанием заменяемого выражения в угловых скобках —  $x_{\langle A \rangle}$ .

При этом подстановка  $A[x := B]$  раскрывается так:

$$A[x := B] = \begin{cases} t_{\langle B \rangle}, & A = x \\ y, & A = y, y \neq x \\ \lambda x.P, & A = \lambda x.P \\ \lambda y.(P[x := B]), & A = \lambda y.P, y \neq x \\ (P[x := B])\ (Q[x := B]), & A = P\ Q \end{cases}$$

Здесь  $t$  — некоторая новая отложенная переменная, ранее в выражении не встречавшаяся.

Естественным образом мы можем определить плоское лямбда-выражение для данного выражения, рассматривая каждую переменную вида  $x_{\langle P \rangle}$  как  $P$ .

Тогда шаг редукции с мемоизацией устроен так:

- Выберем редекс  $(\lambda x.A)\ B$  — например, найдём самый левый редекс в плоском лямбда-выражении, соответствующем данному.
- Если  $(\lambda x.A)$  содержит вхождение отложенной подстановки  $y_{\langle P \rangle}$ , в которую входит заменяемая переменная  $x$ , перед редукцией заменим данное вхождение  $y_{\langle P \rangle}$  на  $P$ . Обратите внимание, случай  $\lambda x.A = y_{\langle P \rangle}$  также надо учитывать.
- Все остальные отложенные подстановки в редексе оставим без изменений — рассматриваем, как переменные. Производим редукцию.

- Если редекс целиком находится внутри какой-то отложенной подстановки — редукцию производим во всех отложенных подстановках по той же переменной.

## Формат входных данных

В первой строке приведены числа  $m$  и  $k$  через пробел. Во второй строке дано лямбда-выражение  $\delta_0$  в грамматике из предыдущего задания.

## Формат выходных данных

Выведите формулы  $\delta_0, \delta_k, \delta_{k+2}, \dots, \delta_{k+(n-1)}, \delta_s$ , по формуле на новой строке.

## Примеры

стандартный ввод	стандартный вывод
10 1 (\x.x) z	((\x.x) z) z
100 1 (\x.y) z	((\x.y) z) y
100 1 (\a.\a.b) c	((\a.(\a.b)) c) (\v0.b)
100 1 (\a.\x.a) (x y)	((\a.(\x.a)) (x y)) (\v0.(x y))

## Задача С. Вывод типа в просто-типизированном лямбда-исчислении

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 10 секунд  
Ограничение по памяти: 512 мегабайт

На вход вашей программе дается лямбда-выражение в следующей грамматике:

$$\begin{aligned} \langle \text{Выражение} \rangle &::= [\langle \text{Применение} \rangle] \backslash \langle \text{Переменная} \rangle \text{ '.' } \langle \text{Выражение} \rangle \\ &\quad | \langle \text{Применение} \rangle \\ \langle \text{Применение} \rangle &::= \langle \text{Применение} \rangle \langle \text{Атом} \rangle | \langle \text{Атом} \rangle \\ \langle \text{Атом} \rangle &::= \text{'(' } \langle \text{Выражение} \rangle \text{' )' } | \langle \text{Переменная} \rangle \\ \langle \text{Переменная} \rangle &::= (\text{'a' } \dots \text{'z' }) \{ \text{'a' } \dots \text{'z' } | \text{'0' } \dots \text{'9' } | \text{' ' } \}^* \end{aligned}$$

Аргументы-переменные в применении разделяются пробелом. В остальных случаях пробелы могут отсутствовать. Любые пробелы между нетерминальными символами (кроме пробела, разделяющего аргументы в применении) — а также начальные и конечные пробелы в строке — должны игнорироваться. Символы табуляции и возврата каретки должны трактоваться как пробелы.

Требуется найти наиболее общий тип этого лямбда-выражения и вывести доказательство того, что лямбда-выражение имеет этот тип, а также найти типы свободных переменных, содержащихся в лямбда-выражении, или же сказать, что лямбда-выражение не имеет типа.

В доказательстве вы можете пользоваться следующими правилами:

Правило	Зависимости	Вывод	Дополнительные условия
№1		$\Gamma, x : \sigma \vdash x : \sigma$	$x \notin \text{dom}(\Gamma)$
№2	$\Gamma \vdash M : \sigma \rightarrow \tau, N : \sigma$	$\Gamma \vdash MN : \tau$	
№3	$\Gamma, x : \sigma \vdash M : \tau$	$\Gamma \vdash \lambda x. M : \sigma \rightarrow \tau$	$x \notin \text{dom}(\Gamma)$

### Формат входных данных

В единственной строке входного файла содержится лямбда-выражение в грамматике из условия. Длина выражения не превышает 255 символов.

Гарантируется, что имена всех вложенных абстракций различны, а также имена абстракций не совпадают с именами свободных переменных.

### Формат выходных данных

Если заданное лямбда-выражение не имеет типа, в единственной строке выходного файла должна быть запись «**Expression has no type**».

Иначе в файле должно быть доказательство. В файле должны отсутствовать пустые строки. Строки доказательства должны идти в правильном порядке. Каждый отступ должен представляться с помощью «\*» — символа «\*» (ASCII 42) и трех последовательных пробелов (ASCII 32). В конце каждой строки должно быть описание правила, которое было применено для вывода этой строки. В остальном следуйте формату из примеров.

Выведенный тип должен быть наиболее общим типом для заданного лямбда-выражения.

### Примеры

стандартный ввод
x
стандартный вывод
x : t1  - x : t1 [rule #1]

стандартный ввод
$(\lambda x. x) (\lambda y. y)$
стандартный вывод
$\vdash ((\lambda x. x) (\lambda y. y)) : (t2 \rightarrow t2)$ [rule #2] $*$ $\vdash (\lambda x. x) : ((t2 \rightarrow t2) \rightarrow (t2 \rightarrow t2))$ [rule #3] $*$ $*$ $x : (t2 \rightarrow t2) \vdash x : (t2 \rightarrow t2)$ [rule #1] $*$ $\vdash (\lambda y. y) : (t2 \rightarrow t2)$ [rule #3] $*$ $*$ $y : t2 \vdash y : t2$ [rule #1]
стандартный ввод
$\lambda a. a' a z8'$
стандартный вывод
$a' : (t1 \rightarrow (t4 \rightarrow t5)), z8' : t4 \vdash (\lambda a. ((a' a) z8')) : (t1 \rightarrow t5)$ [rule #3] $*$ $a' : (t1 \rightarrow (t4 \rightarrow t5)), z8' : t4, a : t1 \vdash ((a' a) z8') : t5$ [rule #2] $*$ $*$ $a' : (t1 \rightarrow (t4 \rightarrow t5)), z8' : t4, a : t1 \vdash (a' a) : (t4 \rightarrow t5)$ [rule #2] $*$ $*$ $*$ $a' : (t1 \rightarrow (t4 \rightarrow t5)), z8' : t4, a : t1 \vdash a' : (t1 \rightarrow (t4 \rightarrow t5))$ [rule #1] $*$ $*$ $*$ $a' : (t1 \rightarrow (t4 \rightarrow t5)), z8' : t4, a : t1 \vdash a : t1$ [rule #1] $*$ $*$ $a' : (t1 \rightarrow (t4 \rightarrow t5)), z8' : t4, a : t1 \vdash z8' : t4$ [rule #1]